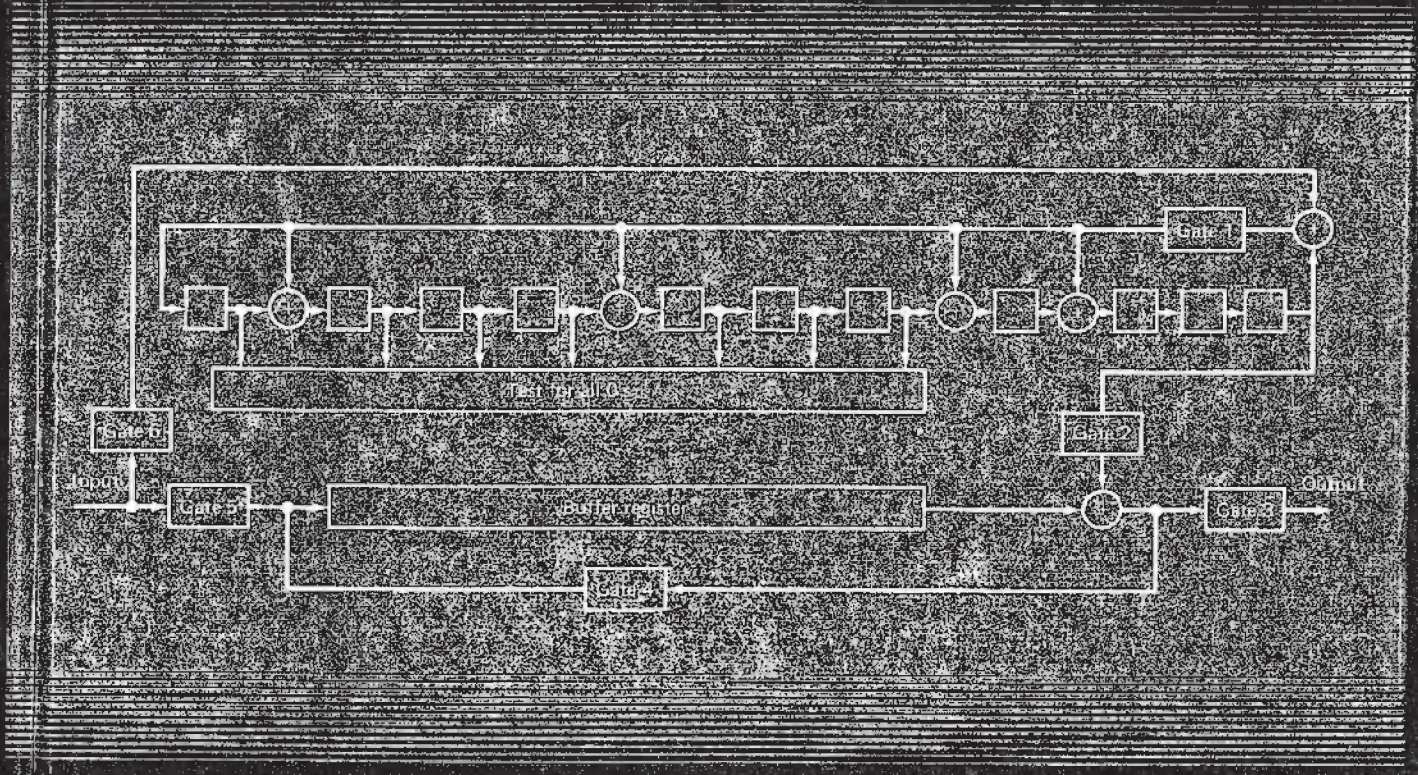
Exhibit K

SHU LIN / DANIEL J. COSTELLO, Jr.

Error Control Coding:

Fundamentals and Applications



PRENTICE-HALL SERIES IN COMPUTER APPLICATIONS IN ELECTRICAL ENGINEERING
Franklin F. Kuo, Editor

ERROR CONTROL CODING

Fundamentals and Applications

SHU LIN
University of Hawaii
Texas A&M University

DANIEL J. COSTELLO, JR.
Illinois Institute of Technology

Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

LIN, SHU.
Error control coding.

(Prentice-Hall computer applications in electrical engineering series)
Includes bibliographical references and index.
1. Error-correcting codes (Information theory)
I. Costello, Daniel J. II. Title.
III. Series.
QA268.L55 001.53'9 82-5255
ISBN 0-13-283796-X AACR2

Editorial/production supervision and interior design by Anne Simpson
Cover design by Marvin Warshaw
Manufacturing buyer: Joyce Levatino

Printed in the United States of America

20 19 18 17 16 15

ISBN 0-13-283796-X

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
EDITORA PRENTICE-HALL DO BRAZIL, LTDA, *Rio de Janeiro*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

3

Linear Block Codes

In this chapter basic concepts of block codes are introduced. For ease of code synthesis and implementation, we restrict our attention to a subclass of the class of all block codes, the *linear block codes*. Since in most present digital computers and digital data communication systems, information is coded in binary digits "0" or "1," we discuss only the linear block codes with symbols from the binary field GF(2). The theory developed for the binary codes can be generalized to codes with symbols from a nonbinary field in a straightforward manner.

First, linear block codes are defined and described in terms of *generator* and *parity-check* matrices. The parity-check equations for a *systematic* code are derived. Encoding of linear block codes is discussed. In Section 3.2 the concept of *syndrome* is introduced. The use of syndrome for error detection and correction is discussed. In Sections 3.3 and 3.4 we define the *minimum distance* of a block code and show that the random-error-detecting and random-error-correcting capabilities of a code are determined by its minimum distance. Probabilities of a decoding error are discussed. In Section 3.5 the *standard array* and its application to the decoding of linear block codes are presented. A general decoder based on the *syndrome decoding* scheme is given. Finally, we conclude the chapter by presenting a class of single-error-correcting linear codes.

References 1 to 4 contain excellent treatments of linear block codes.

3.1 INTRODUCTION TO LINEAR BLOCK CODES

We assume that the output of an information source is a sequence of binary digits "0" or "1." In block coding, this binary information sequence is segmented into *message* blocks of fixed length; each message block, denoted by $\mathbf{u}$, consists of k

information digits. There are a total of 2^k distinct messages. The encoder, *according to certain rules*, transforms each input message $\mathbf{u}$ into a binary n-tuple $\mathbf{v}$ with $n > k$. This binary n-tuple $\mathbf{v}$ is referred to as the *code word* (or *code vector*) of the message $\mathbf{u}$. Therefore, corresponding to the 2^k possible messages, there are 2^k code words. This set of 2^k code words is called a *block* code. For a block code to be useful, the 2^k code words must be distinct. Therefore, there should be a one-to-one correspondence between a message $\mathbf{u}$ and its code word $\mathbf{v}$.

For a block code with 2^k code words and length n, unless it has a certain special structure, the encoding apparatus would be prohibitively complex for large k and n since it has to store the 2^k code words of length n in a dictionary. Therefore, we must restrict our attention to block codes that can be mechanized in a practical manner. A desirable structure for a block code to possess is the *linearity*. With this structure in a block code, the encoding complexity will be greatly reduced, as we will see.

Definition 3.1. A block code of length n and 2^k code words is called a *linear* (n, k) code if and only if its 2^k code words form a k-dimensional subspace of the vector space of all the n-tuples over the field GF(2).

In fact, a binary block code is linear if and only if the modulo-2 sum of two code words is also a code word. The block code given in Table 3.1 is a (7, 4) linear code. One can easily check that the sum of any two code words in this code is also a code word.

Since an (n, k) linear code C is a k-dimensional subspace of the vector space V_n of all the binary n-tuples, it is possible to find k linearly independent code words,

TABLE 3.1 LINEAR BLOCK CODE WITH $k = 4$ AND $n = 7$

Messages	Code words
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 0 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

$\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}$ in C such that every code word $\mathbf{v}$ in C is a linear combination of these k code words, that is,

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1}, \tag{3.1}$$

where $u_i = 0$ or 1 for $0 \leq i < k$. Let us arrange these k linearly independent code words as the rows of a $k \times n$ matrix as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}, \tag{3.2}$$

where $\mathbf{g}_i = (g_{i0}, g_{i1}, \ldots, g_{i,n-1})$ for $0 \leq i < k$. If $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded, the corresponding code word can be given as follows:

$$\begin{aligned} \mathbf{v} &= \mathbf{u} \cdot \mathbf{G} \\ &= (u_0, u_1, \ldots, u_{k-1}) \cdot \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{bmatrix} \\ &= u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \cdots + u_{k-1}\mathbf{g}_{k-1}. \end{aligned} \tag{3.3}$$

Clearly, the rows of $\mathbf{G}$ *generate* (or *span*) the (n, k) linear code C. For this reason, the matrix $\mathbf{G}$ is called a *generator* matrix for C. Note that any k linearly independent code words of an (n, k) linear code can be used to form a generator matrix for the code. It follows from (3.3) that an (n, k) linear code is completely specified by the k rows of a generator matrix $\mathbf{G}$. Therefore, the encoder has only to store the k rows of $\mathbf{G}$ and to form a linear combination of these k rows based on the input message $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$.

Example 3.1

The (7, 4) linear code given in Table 3.1 has the following matrix as a generator matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

If $\mathbf{u} = (1\ \ 1\ \ 0\ \ 1)$ is the message to be encoded, its corresponding code word, according to (3.3), would be

$$\begin{aligned} \mathbf{v} &= 1 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1 + 0 \cdot \mathbf{g}_2 + 1 \cdot \mathbf{g}_3 \\ &= (1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0) + (0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0) + (1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 1) \\ &= (0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1). \end{aligned}$$

A desirable property for a linear block code to possess is the *systematic structure* of the code words as shown in Figure 3.1, where a code word is divided into two parts, the message part and the redundant checking part. The message part consists of k unaltered information (or message) digits and the redundant checking part consists of $n - k$ *parity-check* digits, which are *linear sums* of the information digits. A linear block code with this structure is referred to as a *linear systematic block code*. The (7, 4) code given in Table 3.1 is a linear systematic block code, the rightmost four digits of each code word are identical to the corresponding information digits.

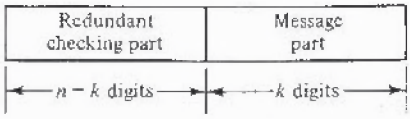


Figure 3.1 Systematic format of a code word.

A linear systematic (n, k) code is completely specified by a $k \times n$ matrix $\mathbf{G}$ of the following form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \left[\begin{array}{cccc|ccccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ p_{20} & p_{21} & \cdots & p_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ & & & & & & & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{array}\right], \tag{3.4}$$

P matrix; $k \times k$ identity matrix

where $p_{ij} = 0$ or 1. Let $\mathbf{I}_k$ denote the $k \times k$ identity matrix. Then $\mathbf{G} = [\mathbf{P}\ \mathbf{I}_k]$. Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded. The corresponding code word is

$$\begin{aligned} \mathbf{v} &= (v_0, v_1, v_2, \ldots, v_{n-1}) \\ &= (u_0, u_1, \ldots, u_{k-1}) \cdot \mathbf{G}. \end{aligned} \tag{3.5}$$

It follows from (3.4) and (3.5) that the components of $\mathbf{v}$ are

$$v_{n-k+i} = u_i \qquad \text{for } 0 \leq i < k \tag{3.6a}$$

and

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} \tag{3.6b}$$

for $0 \leq j < n - k$. Equation (3.6a) shows that the rightmost k digits of a code word $\mathbf{v}$ are identical to the information digits $u_0, u_1, \ldots, u_{k-1}$ to be encoded, and (3.6b) shows that the leftmost $n - k$ redundant digits are linear sums of the information digits. The $n - k$ equations given by (3.6b) are called *parity-check equations* of the code.

Example 3.2

The matrix **G** given in Example 3.1 is in systematic form. Let $\mathbf{u} = (u_0, u_1, u_2, u_3)$ be the message to be encoded and let $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word. Then

$$\mathbf{v} = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By matrix multiplication, we obtain the following digits of the code word **v**:

$$\begin{aligned} v_6 &= u_3 \\ v_5 &= u_2 \\ v_4 &= u_1 \\ v_3 &= u_0 \\ v_2 &= u_1 + u_2 + u_3 \\ v_1 &= u_0 + u_1 + u_2 \\ v_0 &= u_0 + u_2 + u_3. \end{aligned}$$

The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1).

There is another useful matrix associated with every linear block code. As stated in Chapter 2, for any $k \times n$ matrix **G** with k linearly independent rows, there exists an $(n - k) \times n$ matrix **H** with $n - k$ linearly independent rows such that any vector in the row space of **G** is orthogonal to the rows of **H** and any vector that is orthogonal to the rows of **H** is in the row space of **G**. Hence, we can describe the (n, k) linear code generated by **G** in an alternate way as follows: *An n-tuple* **v** *is a code word in the code generated by* **G** *if and only if* $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. This matrix **H** is called a *parity-check matrix* of the code. The 2^{n-k} linear combinations of the rows of matrix **H** form an $(n, n - k)$ linear code C_d. This code is the null space of the (n, k) linear code C generated by matrix **G** (i.e., for any $\mathbf{v} \in C$ and any $\mathbf{w} \in C_d$, $\mathbf{v} \cdot \mathbf{w} = 0$). C_d is called the *dual* code of C. Therefore, a parity-check matrix for a linear code C is a generator matrix for its dual code C_d.

If the generator matrix of an (n, k) linear code is in the systematic form of (3.4), the parity-check matrix may take the following form:

$$\begin{aligned} \mathbf{H} &= [\mathbf{I}_{n-k} \quad \mathbf{P}^T] \\ &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}, \end{aligned} \tag{3.7}$$

where $\mathbf{P}^T$ is the transpose of the matrix $\mathbf{P}$. Let $\mathbf{h}_j$ be the jth row of $\mathbf{H}$. We can check readily that the inner product of the ith row of $\mathbf{G}$ given by (3.4) and the jth row of $\mathbf{H}$ given by (3.7) is

$$\mathbf{g}_i \cdot \mathbf{h}_j = p_{ij} + p_{ij} = 0$$

for $0 \leq i < k$ and $0 \leq j < n - k$. This implies that $\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}$. Also, the $n - k$ rows of $\mathbf{H}$ are linearly independent. Therefore, the $\mathbf{H}$ matrix of (3.7) is a parity-check matrix of the (n, k) linear code generated by the matrix $\mathbf{G}$ of (3.4).

The parity-check equations given by (3.6b) can also be obtained from the parity-check matrix $\mathbf{H}$ of (3.7). Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ be the message to be encoded. In systematic form the corresponding code word would be

$$\mathbf{v} = (v_0, v_1, \ldots, v_{n-k-1}, u_0, u_1, \ldots, u_{k-1}).$$

Using the fact that $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$, we obtain

$$v_j + u_0 p_{0j} + u_1 p_{1j} + \cdots + u_{k-1} p_{k-1,j} = 0 \tag{3.8}$$

for $0 \leq j < n - k$. Rearranging the equations of (3.8), we obtain the same parity-check equations of (3.6b). Therefore, an (n, k) linear code is completely specified by its parity-check matrix.

Example 3.3

Consider the generator matrix of a (7, 4) linear code given in Example 3.1. The corresponding parity-check matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

At this point, let us summarize the foregoing results: For any (n, k) linear block code C, there exists a $k \times n$ matrix $\mathbf{G}$ whose row space gives C. Furthermore, there exists an $(n - k) \times n$ matrix $\mathbf{H}$ such that an n-tuple $\mathbf{v}$ is a code word in C if and only if $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$. If $\mathbf{G}$ is of the form given by (3.4), then $\mathbf{H}$ may take the form given by (3.7), and vice versa.

Based on the equations of (3.6a) and (3.6b), the encoding circuit for an (n, k) linear systematic code can be implemented easily. The encoding circuit is shown in Figure 3.2, where $\rightarrow\boxed{\ }\rightarrow$ denotes a shift-register stage (e.g., a flip-flop), $\oplus$ denotes a modulo-2 adder, and $\rightarrow(p_{ij})\rightarrow$ denotes a connection if $p_{ij} = 1$ and no connection if $p_{ij} = 0$. The encoding operation is very simple. The message $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ to be encoded is shifted into the message register and simultaneously into the channel. As soon as the entire message has entered the message register, the $n - k$ parity-check digits are formed at the outputs of the $n - k$ modulo-2 adders. These parity-check digits are then serialized and shifted into the channel. We see that the complexity of the encoding circuit is linearly proportional to the block length of the code. The encoding circuit for the (7, 4) code given in Table 3.1 is shown in Figure 3.3, where the connection is based on the parity-check equations given in Example 3.2.

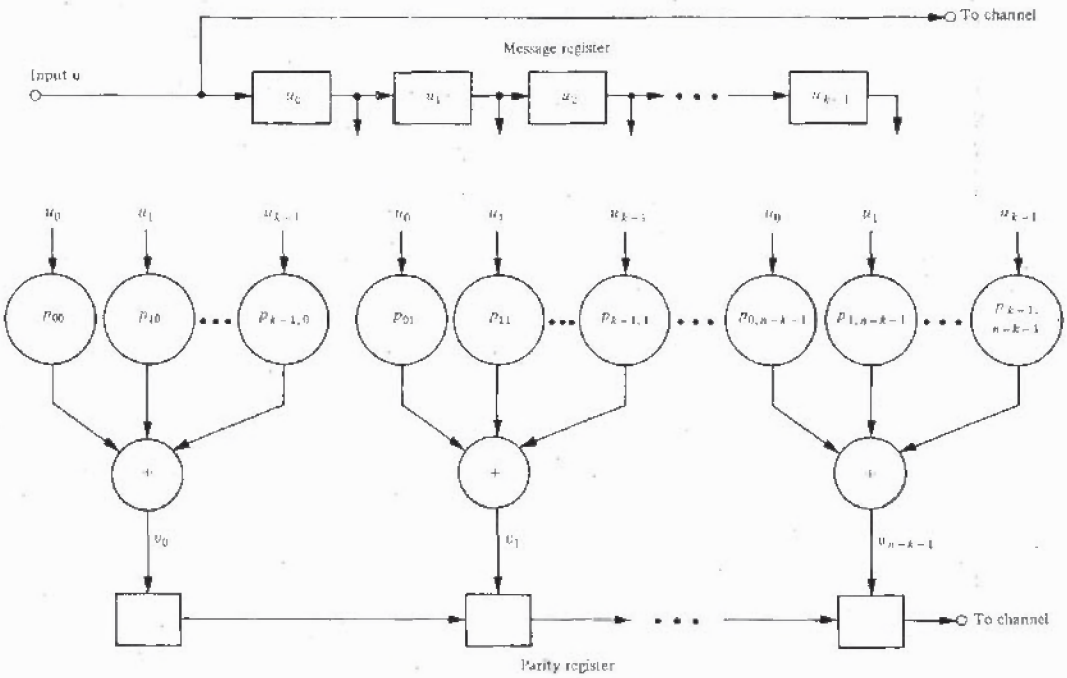


Figure 3.2 Encoding circuit for a linear systematic (n, k) code.

57

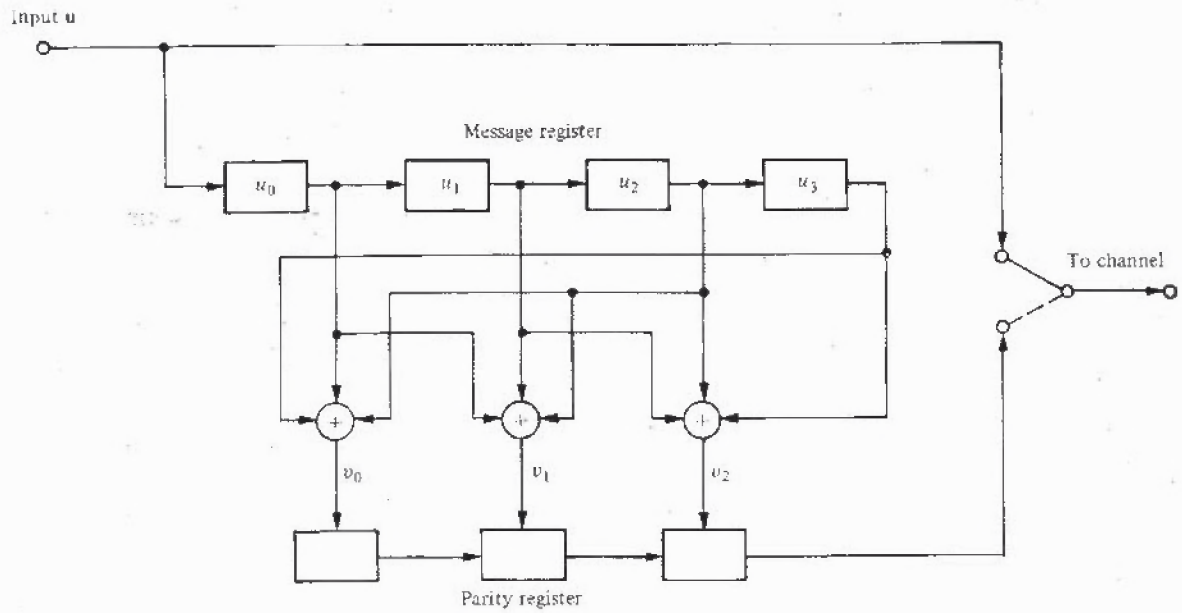


Figure 3.3 Encoding circuit for the (7, 4) systematic code given in Table 3.1.

3.2 SYNDROME AND ERROR DETECTION

Consider an (n, k) linear code with generator matrix $\mathbf{G}$ and parity-check matrix $\mathbf{H}$. Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a code word that was transmitted over a noisy channel. Let $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$ be the received vector at the output of the channel. Because of the channel noise, $\mathbf{r}$ may be different from $\mathbf{v}$. The vector sum

$$\begin{aligned} \mathbf{e} &= \mathbf{r} + \mathbf{v} \\ &= (e_0, e_1, \ldots, e_{n-1}) \end{aligned} \tag{3.9}$$

is an n-tuple where $e_i = 1$ for $r_i \neq v_i$ and $e_i = 0$ for $r_i = v_i$. This n-tuple is called the *error vector* (or *error pattern*). The 1's in $\mathbf{e}$ are the *transmission errors* caused by the channel noise. It follows from (3.9) that the received vector $\mathbf{r}$ is the vector sum of the transmitted code word and the error vector, that is,

$$\mathbf{r} = \mathbf{v} + \mathbf{e}.$$

Of course, the receiver does not know either $\mathbf{v}$ or $\mathbf{e}$. Upon receiving $\mathbf{r}$, the decoder must first determine whether $\mathbf{r}$ contains transmission errors. If the presence of errors is detected, the decoder will either take actions to locate the errors and correct them (FEC) or request for a retransmission of $\mathbf{v}$(ARQ).

When $\mathbf{r}$ is received, the decoder computes the following $(n - k)$-tuple:

$$\begin{aligned} \mathbf{s} &= \mathbf{r} \cdot \mathbf{H}^T \\ &= (s_0, s_1, \ldots, s_{n-k-1}), \end{aligned} \tag{3.10}$$

which is called the *syndrome* of $\mathbf{r}$. Then $\mathbf{s} = \mathbf{0}$ if and only if $\mathbf{r}$ is a code word, and $\mathbf{s} \neq \mathbf{0}$